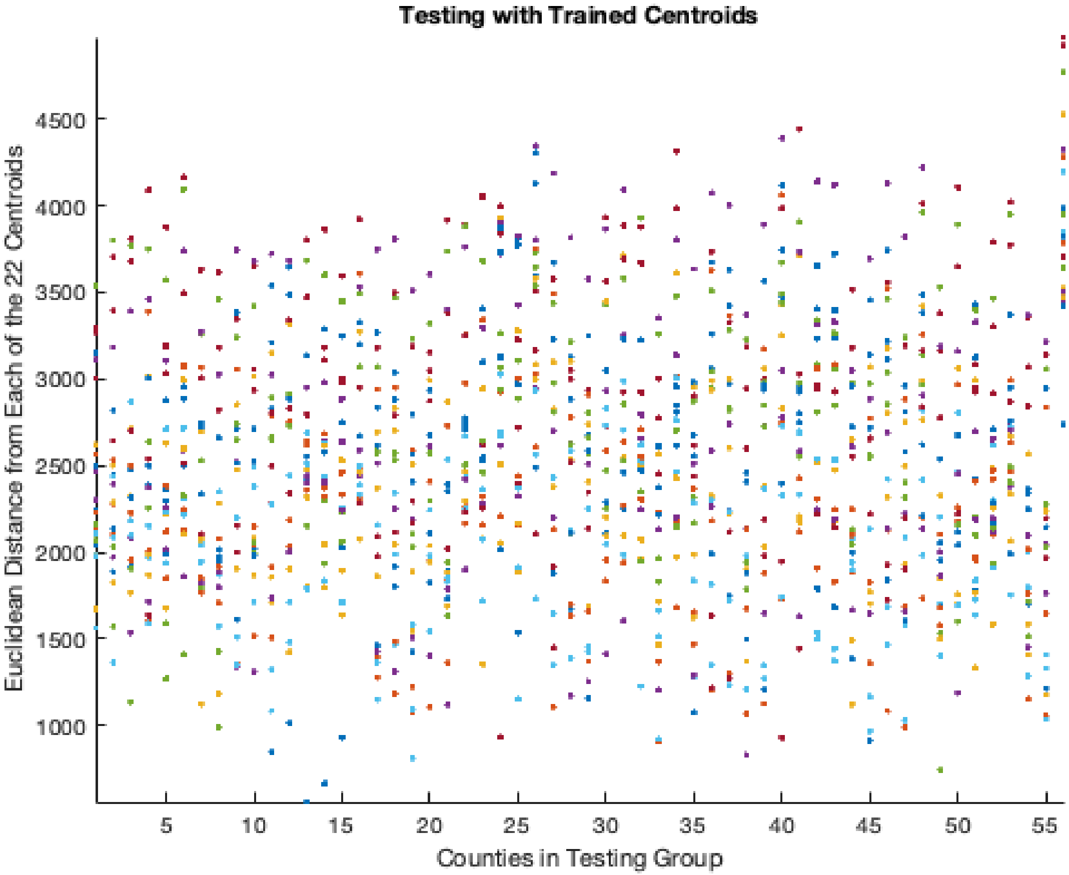
```
nearestNeighbor = [];
centroidTesting = [];

for i = 1:height(centroids)
    for j = 1:height(testingCases)
        % use Euclidean distance to find the nearest centroid
        nearestNeighbor = [nearestNeighbor, norm(centroids(i, :) - ...
            testingCases(j, :))]; % row
    end
    % append each row to matrix
    centroidTesting = [centroidTesting; nearestNeighbor];
    nearestNeighbor = []; % clear current row for the next row of entries
end

figure;
hold on;
for i = 1:height(centroidTesting)
    plot(centroidTesting(i, :), '.');
end
hold off;
axis tight;
title('Testing with Trained Centroids');
xlabel('Counties in Testing Group');
ylabel("Euclidean Distance from Each of the " + k + " Centroids");
exportgraphics(gca, 'testing_with_trained_centroids.png');

correct = 0;
incorrect = 0;

for i = 1:width(centroidTesting) % loop through matrix of centroids
    % set initial minimum value for later comparison
    minimum = centroidTesting(1, i);
    index = 1;
    for j = 1:height(centroidTesting)
        if centroidTesting(j, i) < minimum % find minimum value
            minimum = centroidTesting(j, i);
            index = j; % find index at minimum value
        end
    end
    % check if testing data matches centroids from training data
    if centroid_labels(index) == testing.DIVISION(i)
        correct = correct + 1;
    else
        incorrect = incorrect + 1;
    end
end

% calculate accuracy by dividing # correct by total #
accuracy = correct / (correct + incorrect);
disp("Results:")
disp("Accuracy: " + correct + "/" + (correct + incorrect) + " (" + ...
    round(accuracy * 100 * 100) / 100 + "%)");
```

```
disp("Score: " + (correct - k / 2.0));

% output file containg centroids and centroid_labels
save("competition.mat", "centroids", "centroid_labels");
```

```
Results:
Accuracy: 34/56 (60.71%)
Score: 23
```



Published with MATLAB® R2023a